❒   89

Path optimization for robots in a constrained workspace

Doaa Mahmood Badr, Abbas Fadhal Mahdi
Department of Computer Science, Faculty of Computer Science and Mathematics, Kufa University, Alnajaf, Iraq

Article Info	ABSTRACT
	In this work, the classical A* algorithm serves as path planner to generate the optimum path that would avoid collisions and take the start, collisions, and goal as an input and give the optimal path as an output. The work was done in a static environment, so the coordinates of the obstacles are predefined for the planner. The obtained path is just a sequence of points in space, and this path may be considered later the task space and the first step for another sequential operation like mapping from Cartesian space to joint space, topology optimization, dimensional synthesis, etc. The case study was Lab-Volt 5150 manipulator; it is an accurate educational five degree of freedom 5DOF stationary robot driven by five stepper motors.

Corresponding Author:

Doaa Mahmood Badr,
Department of Computer Science,
Faculty of Computer Science and Mathematics,
Kufa University, Alnajaf, Iraq.
Email: Doaam.alsamee@uokufa.edu.iq

1. INTRODUCTION

Over the last decades, robot manipulators have received much of the robot community attention due to its wide applications in industry for many tasks. The most important function in robot motion is to plan its path to the robot reaches a point in the shortest path consuming less time, consequently consuming less power for its actuators. The main advantage of planning is that it enables a robot to achieve complex goals. Path planning is just a geometric operation to describe the motion of a point on a robot called end-effector. Path planning [1, 2] for articulated robotic manipulators is usually more challenging than for mobile robots because of the high-degrees of freedom. In this paper, A* algorithm was introduced as path generator which takes start position, goal position, and obstacles and returns a series of points which define the optimal path traversed between start and goal position in the configuration space. Path planning occurs at the Cartesian-Coordinate level then, transforming to the joint-Coordinate level occurs using the associated inverse Kinematics. Inverse Kinematics [3, 4] of a robot requires the specification of the end effector's orientation, in this study the orientation is assumed to be constant. Path planning in 2D space greatly saves the time because planning in 3D space is very expensive in consuming the time, however, in some cases under simple 3D situations, it can maps 3D problem area to 2D to use traditional A* algorithm. Examples of 2D planning include routing of telephone traffic [5], navigation through a maze, and Layout of printed circuits boards PCB [6].

2. ALGORITHM

The A*_ (pronounced "Ay star") search algorithm [7] is deal with the searching area as a grid collection and generates a path between two given points. On this path, the whole robot is moving in case of mobile robots, or a part of the robot will follow the path in case of stationary robot manipulators. As shown

in Figure 1, each node being chosen according to the relation. Start node moves to the next node which has minimum cost function f(n) where h(n) is the distance between the (n) node which at start of the algorithm be the start node, and any of its successors and g(n) is the distance between this any successor and the target as shown in (1).

$$f(n)=h(n)+g(n) \tag{1}$$

A* algorithm can be summarized as in the Figure 2 which is present a flow chart for the algorithm.
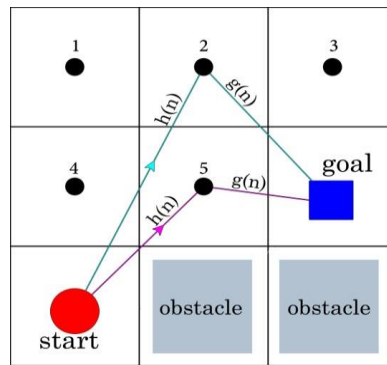

Figure 1. Grid used by A* algorithm to reach the goal

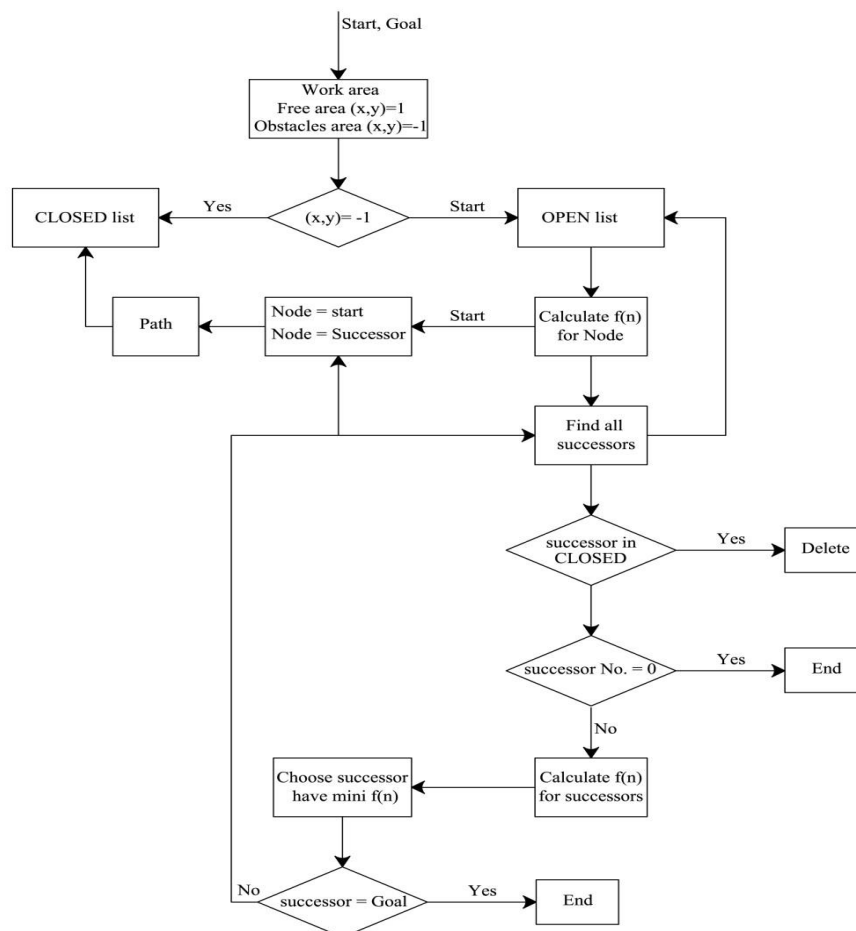Figure 2. A* algorithm flow chart

To force the path to take a specific way, this can be done by throw an obstacles and due to avoidance these obstacle, the path take the desired path in its map as shown in the Figure 3.
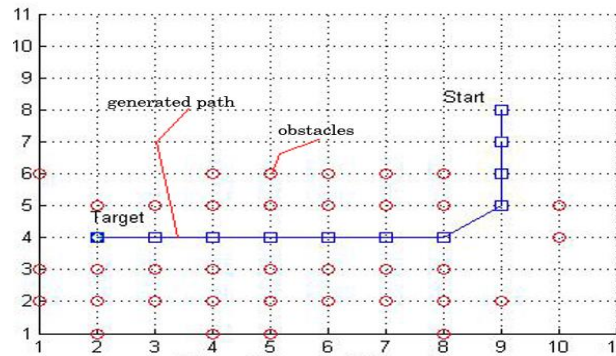

Figure 3. Matlab plot window, forcing a path to take a specific way

3.   ROBOTIC MANIPULATOR SYSTEM

Figure 4 shows a robot manipulator system. Robotic arm can be moved either manually by the teach pendant or by using RoboCim Software which is installed in the interface computer; the software consists of a robot programming language, a program development environment.
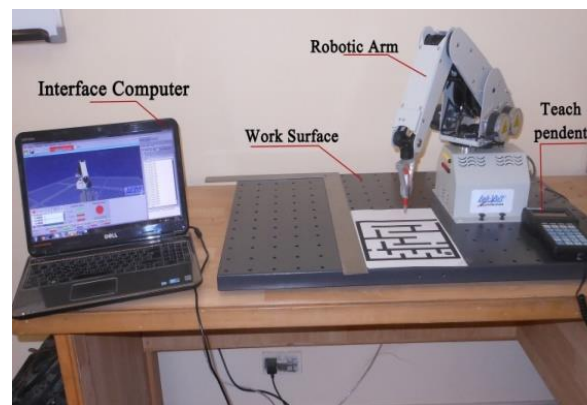

Figure 4. Lab-Volt 5150 manipulator system

4.   PATH PLANNING IN 2D PLAN

Forward and Inverse Kinematics of a Serial links robot manipulators of a revolute joint gives the specification of the end-effector's position and orientation [8]. At the Cartesian-coordinate level, the position and orientation of the end-effector described as a 4x4 homogenous matrix held a 3x3 matrix to describe the orientation and a 3x1 to describe the position. $\begin{bmatrix} 3x3 & 3x1 \\ 0 & 1 \end{bmatrix}$ For the purpose of layout of a PCB we can assume that the orientation and z axis are constants this mean that only x axis and y axis are time varied in the homogenous matrix so the matrix can represented (2),

$$\begin{bmatrix} 3x3 = const & \begin{matrix} x \\ y \\ Z = const \end{matrix} \\ 0 & 1 \end{bmatrix} \tag{2}$$

The constants in the homogenous matrix depend on the operating conditions. To obtain the corresponding angles vector of the manipulator, the homogenous matrix being delivered to the inverse Kinematics equations.

5.   PATH PLANNING IN A FULL OF OBSTACLES AREA

Figure 5 presents a model of an environment full of obstacles designed by AutoCAD package. Then the coordinates of these obstacles defined in A* algorithm and run to generate a path between start point which is its coordinates (120,50) and goal point which is its coordinates (111,262) as shown in Figure 6. The path generated saved in axel file.
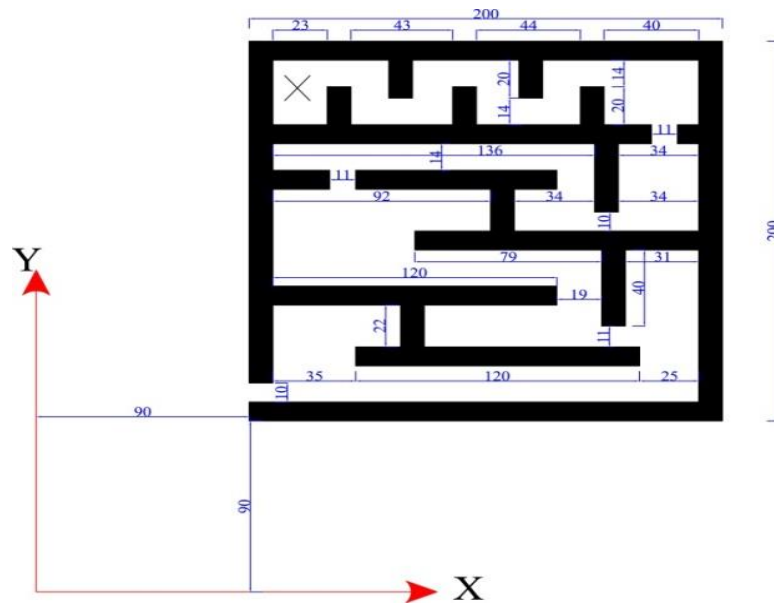

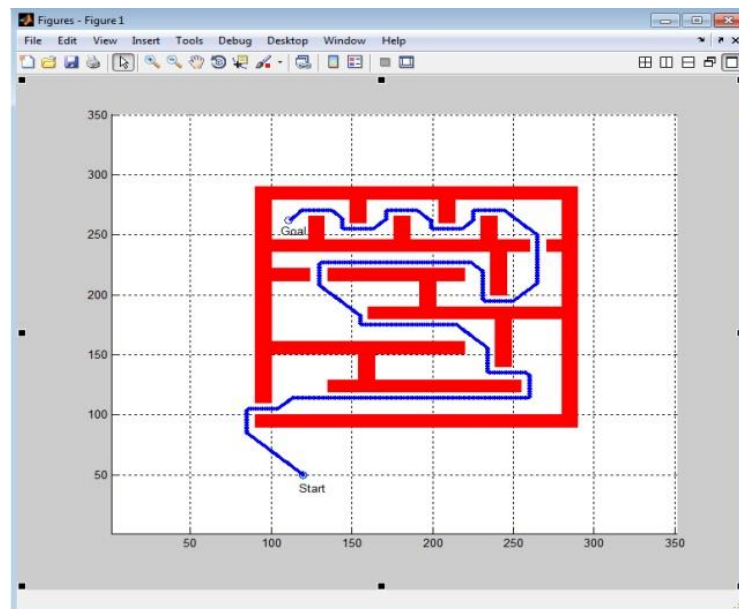Figure 5. Layout of a static environment full of obstacles

Figure 6. Optimum path which avoid all the obstacles in the environment

6.   CONCLUSION

A complete analytical solution to the forward and inverse Kinematics of the 5 D-OF manipulators was driven in this research and all the driven Kinematics governing equations was proved that they are correct and accurate theoretically and experimentally by using a 3D environment developed in C#. Net 2010-XNA 4.0 programming language connected to MATLAB R2014a Software, AutoCAD package

and the resulting data delivered to a real robot manipulator Lab-Volt 5150 and the real arm moved as accurate as the virtual model move.

So, we recommended the driven forward and inverse Kinematics to use in other applications, also we recommend the robot arm to use for educational purposes. A*algorithm has shown an accepted efficiency and accepted work- time for path planning in the static-obstacles environment, and the collision avoidance by this algorithm is very good, and for more accurate results it is recommended that the work-space is divided into a smaller grid in size.

Provide a statement that what is expected, as stated in the "Introduction" chapter can ultimately result in "Results and Discussion" chapter, so there is compatibility. Moreover, it can also be added the prospect of the development of research results and application prospects of further studies into the next (based on result and discussion).

REFERENCES
[1]   Bohlin Robert, "Robot path planning," Chalmers University of Technology, 2002
[2]   Tu Jianping and Yang Simon X, "Genetic algorithm based path planning for a mobile robot, Robotics and Automation," *Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages=1221—1226. (2003).
[3]   Buss Samuel R, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, volume1 7, number 1-19, pages 16, 2004.
[4]   Grochow Keith and Martin Steven L and Hertzmann Aaron and Popovi'c Zoran, Style-based inverse kinematics," ACM transactions on graphics (TOG), volume 23, number3, pages522-531, ACM. 2004.
[5]   Medhi Deep and Ramasamy Karthik, book: "Network routing: algorithms, protocols, and architectures," Morgan Kaufmann, 2017.
[6]   He Tian and Li Debiao and Yoon Sang Won, "A multi-phase planning heuristic for a dual-delivery SMT placement machine optimization," *Robotics and Computer-Integrated Manufacturing*, volume 47, pages 85-94, 2017.
[7]   Kesheng wang, "Efficient inverse position transformation for TR 4000S robot manipulator, IVSL, journal of modeling," *Identification and control*, VOL 10, 1989.
[8]   Lars blackmore and brian Williams, "optimal manipulator path planning with obstacles using Disjunctive programming," *American control conference*, 14-16 June. 2006.